**USER MANUAL**

**COMPUTATION DAG GENERATOR**

# 1.0 GENERAL INFORMATION

General Information section explains in general terms the system and the purpose for which it is intended.

# 1.1 System Overview

The computation DAG generator is a web application. It provide the ability of receiving the computation DAG of an input recursive or multithreaded recursive algorithm. Furthermore user can check whether the input algorithm is recursive or multithreaded recursive.

# 1.2 Organization of the Manual

The user’s manual consists of three sections: General Information, System Summary and Getting Started. General Information section explains in general terms the system and the purpose for which it is intended. System Summary section provides a general overview of the system. The summary outlines the uses of the system’s hardware and software requirements, system’s configuration, user access levels and system’s behavior in case of any contingencies. Getting Started section explains how to get the web application and use the functionality of its.

# 2.0 SYSTEM SUMMARY

System Summary section provides a general overview of the system. The summary outlines the uses of the system’s hardware and software requirements, system’s configuration, user access levels and system’s behavior in case of any contingencies.

# 2.1 System Configuration

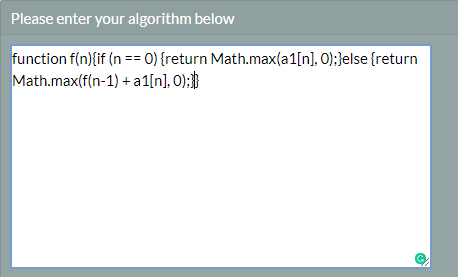
The computation DAG generator operates on any web browser. The application requires connection to internet for the system functionality. After accessing the web site any user can easily user the functionalities as there is no any logging necessary.

# 2.2 User Access Levels

Any user who can access internet can use the application.

# 3.0 GETTING STARTED

* Visit the web application. The URL is given below.
  + <https://computationdaggenerator.000webhostapp.com/>
* In the “Please enter your algorithm below” section, user has to enter an algorithm written in JavaScript. (The algorithm can be recursive or multithreaded recursive.)
* Sample input code is given below
  + function f(n){if (n == 0) {return Math.max(a1[n], 0);}else {return Math.max(f(n-1) + 1[n],0);}}



*Input algorithm structure*

* As shown in the previous section function name should be “f”. The algorithm should be written in JavaScript. Brackets should be used properly.
* In the “Enter the variable value:” section, user have to enter the variable value. Example is given below



*If there is an only one variable*

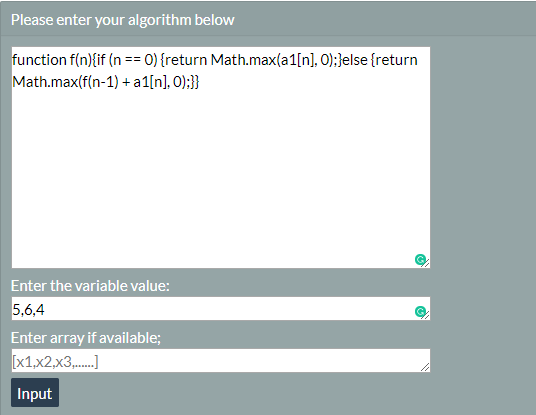


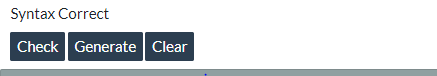
*If there are more than one variable*

* If there is any variable array, input it in the “Enter array if available;” section.

*Input array section*

* After inserting these values select ‘Input’ button.



* If the syntax of the input algorithm is correct, ‘Syntax is correct’ message will display.
* Then user can select any one option from the displayed three options. (Check, Generate, Clear).

*Options for the user*

* Clear option will refresh the browser.
* Generate option will generate the computation DAG of the input algorithm.
* Check option will output whether the algorithm is recursive or multithreaded recursive.